Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

A SIMPLE SOLUTION FOR OPTIMIZING WEEKLY AGENT SCHEDULING IN A MULTI-SKILL MULTI-CHANNEL CONTACT CENTER

Siqiao Li
Ger Koole

Department Of Mathematics
Vrije Universiteit Amsterdam
De Boelelaan 1105
Amsterdam, 1181HV, THE NETHERLANDS

Oualid Jouini

Laboratoire Genie Industriele
CentraleSupélec, Université Paris-Saclay,
9 rue Joliot Curie,
Gif-sur-Yvette, 91190, FRANCE

ABSTRACT

We study the staffing and shift scheduling problem in a multi-skill multi-channel contact center, containing calls, emails and chats. Due to the fact that each channel has its own operating characteristics, the existing solutions developed for multi-skill call centers are not applicable to our problem. In this paper, we first build a high fidelity simulation model at a weekly level to evaluate various Quality of Service (QoS) measurements for a given schedule. Then we propose a simulation-based optimization algorithm to solve the staffing and shift scheduling problem integrally to minimize the total costs of agents under certain QoS requirements. In the numerical experiments, we show the effectiveness of the proposed approach with realistic instances.

1 INTRODUCTION

A contact center as one of the classical service systems, often functions as the first point of contact for customer inquiries and complaints. Nowadays, the modern contact center not only solves customer issues, but also plays significant roles in sales and marketing. According to a global strategic business report (Global Industry Analysts, Inc. 2019), the contact center industry is estimated to have generated $200 billion in revenue worldwide in 2017 and is expected to reach over $400 billion by 2022.

As technology progresses, making telephone calls is not the only way to reach a contact center any more. Other channels, such as email, ticket, and chat are becoming more and more popular. Young customers, for example, prefer chatting over calling since it is convenient. Moreover, emails and chats allow customers to review the dialogue at any time, whereas following instructions during a phone call can be burdensome and tedious. From the operating management point of view, the diversity of channels can help smooth workload, offloading non-urgent services from real-time channels (calls and chats) to non-real-time channels (emails) so that agents can handle them during periods of lower calling load. The chat channel, on the other hand, allows agents to serve several customers simultaneously with slightly longer service time each. However, the operational challenges faced by contact center managers are also more complicated with such a system which has multi-skill agents and customers that arrive from various channels with heterogeneous rates and waiting behaviors. Note that the skill set of an agent may cross the channel.

One of the fundamental challenges faced by contact center managers is how to schedule agents to meet quality of service (QoS) targets at minimum costs. On account of the labor-intensive nature of customer service, in a contact center, almost 60-80% of the operating budget is comprised of the cost of agents. Thus it is crucial to deploy the right number of agents with the right skills to the right schedules (shifts) so as to meet the uncertain, time-varying demands of service. This problem is often called agent scheduling

problem in operations management research. It has been widely discussed in call centers (i.e., with only calls considered), first for single-skill circumstances (Atlason et al. 2004; Ingolfsson et al. 2010) then extended to multi-skill circumstances (Avramidis et al. 2010; Bhulai et al. 2008; Cezik and L'Ecuyer 2008). Although the use of chats and emails is growing, there are few related existing papers. The works considering emails mostly focus on the analysis of call blended policies (Legros et al. 2015; Legros et al. 2018). The staffing decisions and routing policies are addressed for single-skill chat systems, see Cui and Tezcan (2016), Luo and Zhang (2013), and Tezcan and Zhang (2014). As far as we know, no work has studied the scheduling problem considering calls, emails and chats simultaneously (i.e., multi-skill multi-channel contact centers). To fill this gap to some degree, in this paper, we propose a simulation-based optimization approach to solve the staffing and shift scheduling problem integrally for multi-skill multi-channel contact centers. This work is initially motivated by a requirement from a French contact center. Due to the complexity of the system caused by introducing multiple channels, the existing methods developed for call centers are not applicable any more. Therefore, an easy-to-implement approach is strongly needed to help their workforce managers to make agent scheduling decisions.

In a contact center, the arrival rate of each service type is assumed to be constant within a time interval (usually between 15 and 30 minutes). A standard way to solve the agent scheduling problem is considering each interval independently and solve the staffing problem and the shift scheduling problem separately in two steps: 1) the optimal staffing to meet certain QoS requirements is calculated for each interval; 2) given the staffing requirements, the shift scheduling problem is solved to minimize the over-staffing or under-staffing.

Based on this two-step framework, several efficient solutions are proposed (Atlason et al. 2008; Bhulai et al. 2008), however, solving the staffing and shift scheduling problems separately could yield highly sub-optimal solutions in multi-skill systems for a number of reasons. Firstly, in practice, the QoS requirements are usually set for a sequence of many intervals (e.g., a day or a week), so optimizing staffing independently for each interval can easily lead to over-staffing (Avramidis et al. 2010). Secondly, unlike in single-skill systems, the QoS of a service type is not only affected by the number of agents who can handle this service type but also by the configuration of other multi-skill agents. For each interval, similar QoS performance can be derived by various configurations of multi-skill agents. Therefore, we should decide the staffing levels with the best configurations over the planning horizon, taken shift patterns into consideration. These two problems are amplified when we have non-real-time channels such as emails, where interval-independent approaches fail because emails are not necessary to be handled within the interval they arrive. Sometimes they are even allowed to be handled the next day/week. As a result, we have to consider the staffing and shift scheduling problem integrally in the multi-skill multi-channel case, although it increases the complexity of the model.

The first difficulty of the agent scheduling problem is the lack of closed-form expressions for QoS measurements. To cope with this, some papers seek for some approximation methods based on queueing analysis, for example, Pot et al. (2008) proposed an approximation method to evaluate the service level of each service type based on the blocking model, and Cui and Tezcan (2016) gave an approximation based on fluid models to evaluate single-skill chat systems. Simulation is also commonly used (Avramidis et al. 2010; Cezik and L'Ecuyer 2008; Fukunaga et al. 2002). In this paper, we use simulation to evaluate various QoS measurements for a given schedule. One of the reasons is that queueing analysis becomes too complicated when there are more than one channel considered. This is because each channel has its own operating requirements, asking for a different mathematical model to analyze. The details of how calls, chats, and emails are handled in a multi-skill multi-channel contact center are explained in Section 2. Simulation then can work with more complicated systems. Moreover, it needs fewer assumptions and allows us to evaluate various QoS measurements at a time which are very important for practice using. It also takes account of the transience effects between consecutive intervals, which would introduce errors if neglected (Ingolfsson et al. 2007). The main drawback of simulation is that it takes long computation time to derive reliable outcomes. However, in a contact center, agent scheduling decisions are usually

made every week considering weekly shift patterns since agents often have weekly contract hours (e.g., 40 hours). Thus it is acceptable for managers to wait several minutes or even several hours for a good schedule.

Another difficulty is how to solve the staffing and shift scheduling problem integrally. In the existing papers that consider the integral problem, they first model the problem as an integer mathematical programming (Avramidis et al. 2010; Bodur and Luedtke 2017), and solve it with some approximation methods (e.g., cut generation, branch-and-cut, etc.). However, these methods highly depend on the structure of solution space. Introducing emails and chats can easily break it. Instead, we propose a simulation-based local search approach. Considering that we want to solve the agent scheduling problem at a weekly level, general local search algorithms such as tabu search and simulated annealing can be too slow. To speed up, we first derive an initial solution by solving a linear programming model, then apply a heuristic-based local search algorithm to find a local optimal schedule to reach given QoS requirements. To avoid the solution getting stuck in a poor local optimum, we add a permutation step in the end.

The contributions of our work contain: 1) building a simulation model to evaluate various QoS measurements for multi-skill multi-channel contact centers; 2) solving the staffing and shift scheduling problem integrally at a weekly level by proposing an efficient simulation-based heuristic approach.

2 PROBLEM DESCRIPTION

In this section, we first give a high-level description on how a multi-skill multi-channel contact center functions. Thereafter, we formulate the integral staffing and scheduling problem mathematically as an optimization problem.

2.1 Modern Contact Centers

A contact center consists mostly of agents, whose major responsibility is handling customers from different channels. Within each channel, customers can be further categorized to different service types according to their requirements, and each service type corresponds to a skill of agents. For example, many contact centers support different languages (e.g., French, English, etc.), and a customer can choose the channel and language he/she prefers to use. Multi-skill agents are capable to handle more than one service type, according to some routing policy. Therefore, compared to single-skill agents, they can be more flexible and efficient. However, they are often more expensive due to training costs.

Agents are working (and paid) by shifts instead of intervals. A typical shift type over a day is 8 hours long with a 30-minutes break in between. As we mentioned before, in reality, agents are usually scheduled at a weekly level based on their contract hours. Different shift types according to the length, start/end time (day/night shifts), and other variables (include weekend or not) can also have different costs.

Calls: when a call arrives, some routing technology will direct the call to an available agent (if any) who has the skill to handle this call. However, it is often the case that no agents are available, then the call will be held in a queue. The waiting time is usually unknown in advance, and the customer may abandon the queue at any time without receiving service. The length of time a customer is willing to wait before abandonment is called patience, which is usually no more than a few minutes. Once connected to an agent, the customer will be served until his/her problem is resolved and the time it takes to do so is called handling time. While the call is being handled, an interruption by another call with a higher priority may happen. It diverts the agent to the interrupting call, forcing the previous customer to hold until the agent has finished handling the interrupting call. However, this situation is not preferred/allowed by most contact centers since it can lead to long handling times or abandonment. When the call is completed, the agent becomes available again and will handle the next routed call. The waiting time of a call is the time spent in the queue until it is connected to an agent, and it plays a major role in the QoS indicators. For example, one of the most commonly used QoS indicators is the service level (SL), which is defined as

the proportion of customers who wait less than a given time threshold among all arrived customers over a time period. The given time threshold is also known as the acceptable waiting time (AWT).

Chats: similar to calls, chats are also real-time service. However, unlike calls, chats allow agents to handle several customers simultaneously. It is because a customer also needs time to read and type in the reply, and this time can be used by the agent to respond to other customers (if any). Although the chat channel can be more efficient, the handling time of ongoing chats can become longer when a new chat request is accepted. In practice, the maximum number of parallel chats is limited to a number (e.g., 2 or 3) such that the service time distributions will not be negatively affected by the level of concurrency. Interruptions are also not preferred/allowed in the chat channel for the same reason in the call channel. The waiting time of a chat is also defined as the time spent in the queue until it got handled.

Emails: in contrast to the other two channels, emails are not necessarily answered in real time since customers do not abandon. We can consider the patience of an email customer to be very long. Thus the AWT of emails is much longer than calls and chats. Moreover, emails tend to have lower priorities than the other channels and are often allowed to be interrupted. When emails are not handled within the day of arrival, backlogs are generated, which is an important measurement of QoS for emails. The waiting time of an email differs from the other two channels, also includes the handling time since the customer must wait until receiving the reply.

2.2 Mathematical Model

In our problem, we consider a contact center where arriving customers are categorized into I types of service, denoted by i. These customers can be calls, emails or chats that are served by agents with some given routing policy. Agents are divided into G groups, denoted by g, based on the skill set they possessed. We let a binary parameter $Y_{g,i} = 1$ to represent service type i can be handled by agent group g. To distinguish the costs of agent groups, we introduce c_g^A, which can be interpreted as the payment of an agent of group g working on the basic weekly shift (i.e., the cheapest shift).

There are K different weekly shifts considered with the corresponding costs: c_k^S. We let the basic (cheapest) shifts have the cost $c_k^S = 1$. Other shifts according to length, start/end time, and other variables as mentioned earlier, can have higher costs compared to the basic ones, for example, $c_k^S = 1.5$. Note that instead of using the real payment of a shift, we give the relative cost of each shift compared to the basic shifts. Thus by multiplying c_g^A and c_k^S, we can derive the payment of an agent of group g working on shift k. Similar to $Y_{g,i}$, we introduce a binary parameter $X_{k,t}$ to represent the availability of shift k in interval t; $X_{k,t} = 1$ means shift k includes interval s, $X_{k,t} = 0$, otherwise. We let T represents the total number of intervals of the week, denoted by t. For example, if the interval length is 30 minutes, $T = 336$ with 48 intervals daily. As we mentioned earlier, the arrival rate of each service type is considered to be constant within each interval. Therefore, we can use $\lambda_{i,t}$ to represent the estimated arrival rate of service type i in interval t. Note that we do not restrict our model to any particular arrival process, handling time distribution, or patience distribution, instead we only need to be able to simulate them. In practice, the arrival rates are normally derived from the forecast results, and redials and reconnects can also be included. Finally, the schedule can be represented easily by the decision variable $n_{g,k}$, which is the number of agents staffed to group g, and shift k.

The objective of our optimization problem is reaching QoS targets at minimum costs. We choose service level (SL) as the QoS measurement. It is the most commonly used measurement which is defined as the proportion of customers who wait less than a given time threshold (AWT) among all arrived customers over a time period. In practice, managers often pay attention to daily or weekly SLs and set a target to each service type and sometimes a set of several service types. All of them are considered in our problem. The QoS target is often called service level agreement (SLA), and it is the minimum SL a service type (or set) needs to meet. For emails, an additional QoS target related to the backlog is considered. The backlog target suggests the maximum percentage of received emails that can be transferred to the next day (or week). A penalty (cost) is generated proportional to the percent of emails carried over. Other QoS measurements

such as abandonment and occupancy are evaluated using simulation, but they are not considered in the current optimization model.

Now, we give the mathematical model for the integral staffing and shift scheduling problem. The objective function is composed of two parts: the penalty cost caused by breaching the QoS targets C_{QoS} and the agent cost C_{agent}. It can be written as:

$$
\begin{aligned}
&C_{\text{overall}} = C_{\text{QoS}} + C_{\text{agent}},\\
&C_{\text{QoS}} = w_1 C_{\text{SLA}^{\text{week}}} + w_2 C_{\text{SLA}^{\text{day}}} + w_3 C_{\text{backlog}}, &(1)\\
&C_{\text{SLA}^{\text{week}}} = \sum_{i=1}^{I} \mathbb{E}(\max\{\text{SLA}_i^{\text{week}} - \text{SL}_i, 0\}) + \sum_{s=1}^{S} \mathbb{E}(\max\{\text{SLA}_s^{\text{week}} - \text{SL}_s, 0\}), &(2)\\
&C_{\text{SLA}^{\text{day}}} = \sum_{i=1}^{I} \sum_{d=1}^{7} \mathbb{E}(\max\{\text{SLA}_i^{\text{day}} - \text{SL}_{i,d}, 0\}), &(3)\\
&C_{\text{agent}} = \sum_{g=1}^{G} \sum_{k=1}^{K} c_g^A c_k^S n_{g,k},
\end{aligned}
$$

where w_1, w_2 and w_3 are weights to adjust the importance between different targets, s in Equation (2) represents the index of the set of service types, and d in Equation (3) is the index of the day. In reality, the SLA of each day is usually the same. C_{backlog} can be calculated in a similar way as $C_{\text{SLA}^{\text{week}}}$, hence we do not write down all the details for simplicity.

In the end, the optimization problem can be modeled as

$$
\begin{aligned}
\min \quad & C_{\text{QoS}} + C_{\text{agent}} &(4)\\
\text{s.t.} \quad & N_k^l \le \sum_{g=1}^{G} n_{g,k} \le N_k^u, \quad \forall k \in \{1,\dots,K\},\\
& N_g^l \le \sum_{k=1}^{K} n_{g,k} \le N_g^u, \quad \forall g \in \{1,\dots,G\},\\
& n_{g,k} \in \mathbb{N}, \quad \forall k \in \{1,\dots,K\}, g \in \{1,\dots,G\},
\end{aligned}
$$

where N_g^l, N_g^u (N_k^l, N_k^u) correspond to the minimal and maximal number of agents can be scheduled for group g(shift k) respectively. We can set $N_g^l = N_k^l = 0$, and $N_g^u = N_k^u = \infty$ if we do not want to cut out any optimal schedules. However, in practice, these constraints can be necessary when the number of agents could be hired (has been hired) is fixed. To avoid infeasible solutions, instead of setting the QoS targets as hard constraints, we put them into the objective function as penalty costs. Note that the weight of C_{agent} is set to 1 as a reference. In order to ensure that the QoS targets are met, the weights in Equation (1) should be set much higher than the agent costs.

3 SIMULATION SETTING

In this section, we explain how the simulation model is built to evaluate all the QoS measurements needed in the optimization problem. The simulation model is built in C++, to let the simulation as realistic as possible, we take many details of the operation of a contact center into consideration. Similar simulation models that only consider calls have been built in other research. However, this is the first one to deal with a blend of calls, emails and chats. A number of special points need to be treated differently when considering email and chat.

The operation of a contact center is modeled via a discrete-event simulation, where the change of system states is triggered by events, but different states also affect the time at which an event happens.

The input of the simulation model is the setting of the contact center, and for a given schedule the QoS is evaluated for each interval, each day and the whole week.

Specifically, three lists are constructed in the simulation model, representing events, working agents, and queues. All events are stored in the event list sequentially in time. The list is initiated with "arrival" events of all customers generated according to the given arrival distribution per interval. For each service type, the arrival process is assumed as Poisson process with rate $\lambda_{i,t}$, derived from the forecast. The end of each interval is also considered as a clock event and is inserted in the list. This "end-of-interval" event will trigger the change of the working agent list according to the given schedule. When an arrival event happens, the simulation checks whether an agent with the right skill is idle in the current agent list. If an agent is available, the arrival event (customer) is handled and a "departure" event will be inserted into the event list after random handling time. Otherwise, the customer will be added to the queue. For the chat and call customers, an "abandon" event will be inserted after a random amount of patience to the event list. Consequently, a customer will leave the queue if he/she has not been served by the time the abandon event occurs. If there are given redial rates or reconnect rates, a new "arrival" event representing a redial/reconnect will be triggered when an abandon/departure event happens based on the redial/reconnect probability.

We create two agent lists for each agent group: the main list and a chat list. If an agent group has no chat skill, the chat list remains empty. When a chat customer occupies an agent, we will remove the agent from the main list and add to the chat list. Once the agent becomes idle, he/she will be returned back to the main list. When a departure event happens, the formerly occupied agent becomes idle, and then, according to a given routing policy, the first customer from the selected queue will be removed from the queue and occupy the agent. However, if the departure event belongs to a chat customer, we need to check whether there are any other chats are currently handled by the agent. If so, this agent can only seek for another chat customer to serve. One will find that this chat property may lead to the agent has less chance to work on calls or emails, especially when the chat volume is high. Therefore, we let a new arrival chat first search for the agents in the chat list who are already working on chats but not saturated yet. This routing policy is called Most Busy First Policy in a chat system (Legros and Jouini 2019). Figure 1 is given to illustrate the structure of the simulation model intuitively.

During the simulation, all states related to QoS are recorded. In a single simulation run, various QoS measurements are evaluated. The most important ones are listed here. $\mathrm{SL}_{i,t}$: the service level of each service type per interval. As we mentioned earlier, it cannot be measured per interval independently, because emails (or the calls/chats arrive at the end of an interval) are not necessarily to be handled within the interval they arrive. Hence we attach the arrival interval to all the events to trace which interval they belong to. Moreover, the waiting time of an email also includes the handling time, so that needs to be treated differently. The service level for each day and the week can also be evaluated in the same way. Abandonment rate is evaluated only for calls and chats, but we need to evaluate backlogs for emails. Usually, a weekly backlog is considered in reality. An initial backlog from the previous week is added to the queue at the beginning of the simulation.

4 SCHEDULE OPTIMIZATION

Simulation optimization works by intelligently exploring different values of decision variables to find the best objective results for a problem. For the agent scheduling problem, the decision variables are $n_{k,g}$ and the objective function is shown in Equation (4). Given the complexity of this problem, we propose a heuristic-based simulation optimization method as follows to solve the problem.

Step 1: an Integer Linear Programming (ILP) is solved to obtain an initial schedule.

Step 2: a local search algorithm is designed to add agents to one of the agent group/shift combinations until either all SLAs are satisfied or the marginal cost for adding new agents is too high.

Step 3: a number of random shift permutations between different agent groups are carried out to check whether the solution can be further improved.

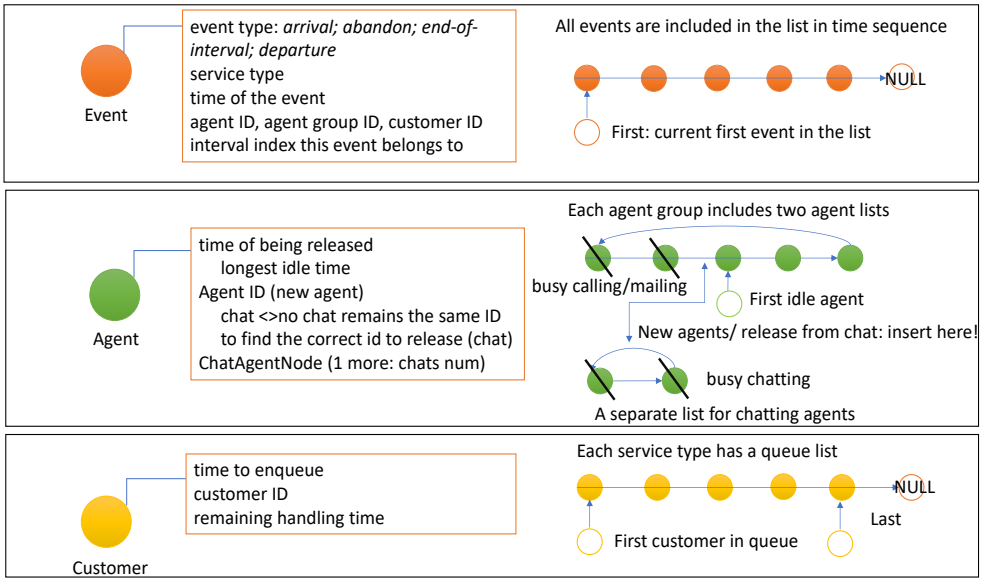

Figure 1: Simulation structure.

Before implementing the algorithm, we first check whether a given contact center can be divided into smaller independent parts based on skill sets of agents. The definition of an independent part here is the agent scheduling of a part will not affect the QoS of any service type in any other part. In other words, there is no skill overlapped between agents from different parts. Therefore, we can optimize each part independently (in parallel). For a small contact center, we can just manually divide it based on the definition. For a large contact center, we can structure an undirected graph based on the skill sets and then use the algorithm developed for connectivity in graph theory. A small example is given in Figure 2. In the following of this section, we explain how the algorithm works in a single part.

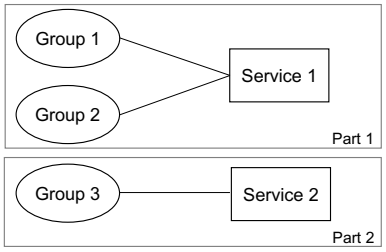

Figure 2: Two independent parts in a contact center.

$$
\begin{aligned}
\min \quad & C_{\text{agent}} \\
\text{s.t.} \quad & \sum_{k=1}^{K} n_{g,k} X_{k,t} = s_{g,t}, \quad \forall g \in \{1,\ldots,G\}, t \in \{1,\ldots,T\}, \\
& \sum_{i \in \mathscr{J}} \alpha_i \rho_{i,t} \leq \sum_{i \in \mathscr{J}} \sum_{g=1}^{G} Y_{g,i} s_{g,t}, \quad \forall \mathscr{J} \subseteq \{1,\ldots,I\}, t \in \{1,\ldots,T\}, \qquad (5) \\
& n_{g,k} \in \mathbb{N}, \quad \forall g \in \{1,\ldots,G\}, k \in \{1,\ldots,K\}, \\
& s_{g,t} \in \mathbb{N}, \quad \forall g \in \{1,\ldots,G\}, t \in \{1,\ldots,T\},
\end{aligned}
$$

As discussed before, simulation often takes long computation time. Thus a good initial solution can help a lot. It also can guide the heuristic algorithm to a better solution. In our case, we relax the service level requirements so that avoid any simulation in the original optimization problem and turn it into an ILP. In this way, no simulation is involved so that the initial solution can be derived quickly. The ILP is modeled as above.

where $s_{g,t}$ is the auxiliary variable introduced as the staffing level of agent group g over interval t; $\rho_{i,t}$ is the required workload (translated to the number of agents) of service type i in interval t, which can be estimated by its arrival rate and average handling time; and the set $\mathscr{J}$ is defined as all the subsets of service types. Therefore Constraint (5) implies that for any subset of service types, the total staffing levels should meet α_i times the overall workload requirements, where α_i is a constant set for each service type. Note that since the workload is interval-based, we should not consider emails in this model. Moreover, the workload should be estimated differently for chats. There are several approximation methods can be used. A simple one can be dividing the total workload by the maximal number of chats an agent can handle at a time.

The optimal solution of the ILP offers a lower bound of the number of agents needed for each agent group/shift combination, minimizing the total cost of agents, which gives an excellent start for the local search algorithm. It can be proved that the optimal solution of the ILP is the minimal cost of agents that keep the system steady. For the sake of the length of the paper, we do not give the proof here. However, the number of constraints of the ILP increases exponentially along with the number of service types because of $\mathscr{J}$, leading to slow calculation. In that case, several alternatives can be used: either further relax the ILP to LP, or use a max-flow-problem-based algorithm (Cezik and L'Ecuyer 2008).

Once we derive the initial schedule, we can start the second step: applying the local search algorithm. In general, a local search algorithm needs four components to structure: an initial schedule, step size, search direction, and stop conditions. In our problem, a schedule M is an integer vector which can be written as:

$$\mathbf{M} = (n_{1,1}, n_{1,2}, ..., n_{1,k}, n_{2,1}, n_{2,2}, ..., n_{2,k}, ..., n_{G,1}, n_{G,2}, ..., n_{G,K}).$$

Naturally, the step size is 1, and the search direction can be represented by (g,k), which is the combination of agent group/shift. Therefore, in each step, we add one agent to one of the group/shift combinations. We stop adding agents if all SLAs are satisfied ($C_{\text{QoS}} = 0$) or the marginal cost for adding new agents is too high.

The details are shown in Algorithm 1, and heuristics are introduced when we decide which agent group/shift combination to add next. Specifically, we calculate a score r_k for each shift k, considering the improvement needed for all the intervals the shift can cover:

$$r_k = \sum_{t=1}^{T} \sum_{i=1}^{I} \max(\Delta \text{SL}_{i,t}, 0) X_{k,s} \lambda_{i,t},$$

where $\Delta \text{SL}_{i,t}$ is the difference between the current service level and the highest SLA among all SLAs that are related to service type i ($\text{SLA}_i^{\text{week}}$, $\text{SLA}_i^{\text{day}}$, $\text{SLA}_s^{\text{week}}$). Then the shift with the highest score will be chosen. However, which agent group to add next is evaluated by simulation, by comparing which agent group/chosen shift combination gives the lowest overall cost. It is because adding an agent with a certain skill set improves not only the SL of the service types included in the skill set but also all other service types. Based on the routing policy, the effects on different service types can be quite different and also very difficult to estimate. Although running simulation for each agent group can be time-consuming, it can offer us a better searching direction.

Due to the randomness in simulation, it may need hundreds of runs to derive reliable SL results, which leads to unacceptable computation time. Instead, we only run a small number of times for each trial (five for instance), implementing Common Random Number (CRN) method, and then use a two-sample t-statistics to compare two schedules. If the two schedules cannot be distinguished, the number of required runs is

estimated, assuming they have equal variance, however, if this number is too big, which implies the two solutions are very close, then the solution with less agent cost will be returned.

Algorithm 1 Local search algorithm for finding near optimal schedules.

initialization: $\mathbf{M}^0 = \mathbf{M}^{init}$, evaluate $C_{\text{overall}}(\mathbf{M}^0)$, $n = 0$
while $C_{\text{Qos}}(M^n) > 0$ or $C_{\text{overall}}(\mathbf{M}^n) \leq C_{\text{overall}}(\mathbf{M}^{n-1})$ **do**
    calculate r_k if $\{k \mid \sum_{g=1}^{G} n_{g,k} \leq N_k^u, k \in \{1,\ldots,K\}\}$
    $k^n = \arg\max_k(r_k)$
    run simulation to estimate $C_{\text{overall}}(\mathbf{M}^n + \vec{e}_{g,k^n})$ if $\{g \mid \sum_{k=1}^{K} n_{g,k} \leq N_g^u, g \in 1,\ldots,G\}$
    $g^n = \arg\min_g(C_{\text{overall}}(\mathbf{M}^n + \vec{e}_{g,k^n}))$
    $\mathbf{M}^{n+1} = \mathbf{M}^n + \vec{e}_{g^n,k^n}$
    $n = n+1$
if $C_{\text{Qos}}(\mathbf{M}^n) == 0$ **then**
    return $\mathbf{M}^n$
else
    return $\mathbf{M}^{n-1}$

The last step is used to check whether the schedule can be further improved. Briefly, We first check whether any of the schedule shift can be removed without breaching any SLA, and then we randomly permute two scheduled shifts to see whether it leads to a better solution.

5 NUMERICAL RESULTS

In this section, we show how the proposed optimization algorithm can be used to optimize agent scheduling. We consider a real case from the contact center, which motivates this work.

5.1 Scenario Setting

There are 7 service types and 5 agent groups in the contact center. The parameters for setting up the agent scheduling problem are in Table 1. The forecasts of each interval (30 minutes) over the week can be found in Figure 3. Both patience and handling time are assumed as an exponential distribution with the given average. In the right side of the table, the skill set of each agent group is represented by $Y_{g,i}$. The email and chat channels only support French. The maximum number of parallel chats is limited to 2. Via analyzing the historical data, handling two chats simultaneously (in this contact center) will not affect the handling time too much. Agents in Group 1 (G1) can communicate in French and G2-G5 have their own language skills plus French. As a result, agents in G2-G5 cost 1.25 times more than G1. The routing policy is static priority policy. In specific, G2-G5 handle French calls with a lower priority and G1 gives the highest priority to calls, then chats, emails in the end.

There are 96 weekly shifts considered in total with the same cost as 1: $C_k^S = 1$ for all k. There are no limits on the number of agents can be assigned.

Recall Equation (4) that our objective is to minimize the cost of failing to meet the SLAs plus the cost of agents. In this case, a weekly SLA of 68% is given for individual service type of calls, and 65% and 80% for emails and chats, respectively. The weekly SLA of both sets (see column Set in Table 1) is given as 74%. We set w_1 and w_2 as 25 times the maximal agent cost among all agent groups, and w_3 to be 1/5 of w_1. These parameters are set according to the importance of different SLAs; however, to make sure the schedule can meet the SLAs, they should be much larger than the agent costs.

5.2 Results

In this subsection, we first verify the proposed algorithm in a single-skill scenario and a multi-skill scenario with only calls, comparing to the revised stationary independent period-by-period (SIPP) model and the

Table 1: Service types setup.

Service	Patience	AHT	AWT	SLA^{week}	Forecast	Set	G1	G2	G3	G4	G5
French	600s	515s	60s	68%	13109	FR	1	1	1	1	1
Italian	600s	505s	60s	68%	835	EU		1			
German	600s	511s	60s	68%	675	EU			1		
Spanish	600s	506s	60s	68%	639	EU				1	
Portuguese	600s	502s	60s	68%	36	EU					1
Email-French	∞	602s	1h	65%	430	FR	1				
Chat-French	600s	640s	90s	80%	4240	FR	1				

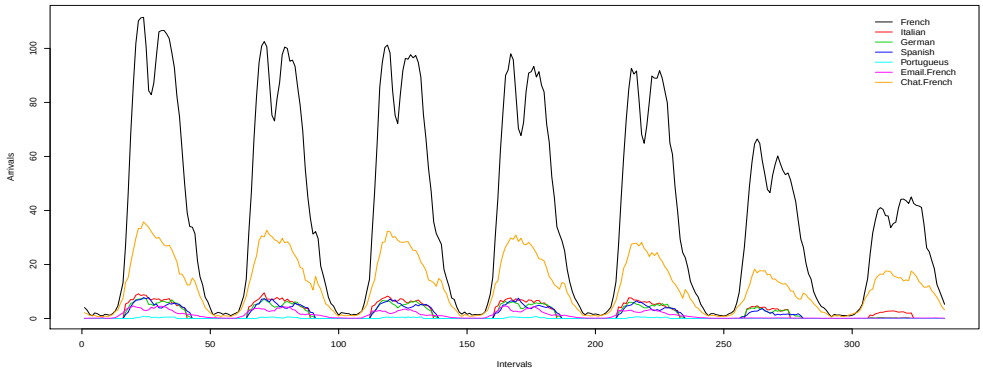

Figure 3: Interval forecast of each service type.

solution proposed by Pot, Bhulai and Koole (Pot et al. 2008; Bhulai et al. 2008). Then it is tested in the multi-channel scenario using the real case above.

5.2.1 Single-skill Scenario & Multi-skill Scenario

The SIPP model is based on the set-covering problem, see the details in Avramidis et al. (2010). In the single-class single-skill scenario, its result is a very good approximation of the optimal solution, as the only error is introduced by ignoring the transience effects. We consider only French calls on Monday in the above case and apply the SIPP model to optimize the agent scheduling. It scheduled 52 agents to meet the SLA (68%). This schedule has likely over-staffed as its SL is 0.71 which is 0.03 higher than the agreement. Using the proposed simulation-based heuristic, we are able to find the optimal schedule with 51 agents and the SL is 68.17%.

For the multi-skill scenario, we consider French, Italian, German and Spanish calls on Monday and apply the proposed two-step based staffing and shift scheduling solution by Pot et al. It scheduled 20 agents (7,4,2,7) to meet the SLA (80%), however, when we test the suggested schedule in the simulation, the SL of Spanish call does not reach the SLA. Our approach only deployed 19 agents (7,4,2,6), which leads to higher service levels, see Table 2 for details. Note that we only consider Monday because the compared approaches do not support weekly optimization.

Table 2: Service level comparison for multi-skill scenario.

Service types	French	Italian	German	Spanish
Sim-base heuristic	83%	84%	83%	85%
Two-step algorithm	82%	86%	86%	72%

5.2.2 Multi-channel Scenario

For the multi-channel setting, there is no other existing solution can be compared with. Therefore, we compare the results with what a workforce manager usually do in reality. We also test the algorithm in different settings to see whether the new requirements are recognized and picked up.

We run the algorithm six times and the best optimal schedule deployed 92 agents with (29,14,17,16,16) corresponding to G1-G5. The average computation time is 4.2 minutes per run, and the objective values do not change much (129.30 - 130.5). The SLs of each service type and set are shown in Table 3.

Table 3: Optimal schedule for the multi-channel case.

	Fre	Ita	Ger	Spa	Por	Email	Chat	FR	EU
Mon	84%	70%	76%	77%	89%	96%	84%	84%	74%
Tue	80%	65%	73%	74%	88%	94%	82%	81%	71%
Wed	83%	74%	81%	77%	93%	94%	79%	82%	77%
Thu	82%	71%	74%	77%	94%	91%	76%	81%	74%
Fri	94%	77%	88%	83%	98%	99%	92%	94%	83%
Sat	91%	85%	73%	73%	-	99%	90%	91%	77%
Sun	46%	42%	27%	-	-	80%	55%	49%	41%
Week	82%	71%	77%	77%	93%	95%	80%	82%	75%

First, we find all the SLAs are met by the schedule, the blank SLs are caused by a closed day. Since we only give weekly SLAs, some of the days may have lower SLs, see Sunday. This is because the customer volume is quite low on Sunday, which can be found in Figure 3. Later, we test the algorithm with additional daily SLAs: 65% for calls and 75% for chats. The obtained schedule deployed 7 more agents but the SLs on Sunday become 80%, 72%, 74%, -, -, 95%, 75% corresponding to each service type.

Second, we find the SLs of the email channel is quite high although its SLA is only 65% and it has the lowest priority. This can be explained by three points: 1) the good property of emails that they can be treated later; 2) the arrival patterns of other real-time channels are similar, thus there are some intervals when agents have no calls/chats to handle; 3) the volume of emails is not high. We then consider a scenario which has an agent group dedicated to emails. Not surprisingly, two more agents are added. For the scenario which has both emails and chats are handled independently, the result is worse with 99 agents scheduled. In reality, due to the lack of optimization tool, the schedule is often obtained by treating different channels separately and sum up in the end. We are glad to find that the proposed algorithm can capture the possible efficiency improvement in between within reasonable computation time.

Considering the length of the paper, we can not show all the results. During the work, we have tested the algorithm in plenty of scenarios and we think implementing our approach can result in a significant reduction in staffing and scheduling costs for complex multi-skill multi-channel contact centers.

6 CONCLUSION

In this paper, we have considered the staffing and shift scheduling problem integrally for multi-skill multi-channel contact centers. To solve it, we first build a realistic simulation model to evaluate various Quality of Service (QoS) measurements for the complex service system with multiple blended channels handled by multi-skill agents. Then a well-structured simulation-based heuristic algorithm is proposed to find the (near) optimal schedule to meet the various QoS requirements. Numerical experiments have shown the effectiveness of the proposed approach. In future work, we want to search for solutions to further speed up the algorithm, and a promising direction can be training a machine learning model based on the simulation results.

REFERENCES

Atlason, J., M. A. Epelman, and S. G. Henderson. 2004. "Call Center Staffing with Simulation and Cutting Plane Methods". *Annals of Operations Research* 127(1-4):333–358.

Atlason, J., M. A. Epelman, and S. G. Henderson. 2008. "Optimizing Call Center Staffing Using Simulation and Analytic Center Cutting-plane Methods". *Management Science* 54(2):295–309.

Avramidis, A. N., W. Chan, M. Gendreau, P. Lecuyer, and O. Pisacane. 2010. "Optimizing Daily Agent Scheduling in a Multiskill Call Center". *European Journal of Operational Research* 200(3):822–832.

Bhulai, S., G. Koole, and A. Pot. 2008. "Simple Methods for Shift Scheduling in Multiskill Call Centers". *Manufacturing & Service Operations Management* 10(3):411–420.

Bodur, M., and J. R. Luedtke. 2017. "Mixed-Integer Rounding Enhanced Benders Decomposition for Multiclass Service-System Staffing and Scheduling with Arrival Rate Uncertainty". *Management Science* 63(7):2073–2091.

Cezik, M. T., and P. L'Ecuyer. 2008. "Staffing Multiskill Call Centers via Linear Programming and Simulation". *Management Science* 54(2):310–323.

Cui, L., and T. Tezcan. 2016. "Approximations for Chat Service Systems Using Many-server Diffusion Limits". *Mathematics of Operations Research* 41(3):775–807.

Fukunaga, A., E. Hamilton, J. Fama, D. Andre, O. Matan, and I. Nourbakhsh. 2002. "Staff Scheduling for Inbound Call and Customer Contact Centers". *AI Magazine* 23(4):30–30.

Global Industry Analysts, Inc. 2019. "Call Centers - A Global Strategic Business Report". Global Industry Analysts, Inc. https://www.strategyr.com/Call_Centers_Market_Report.asp. accessed May 2019.

Ingolfsson, A., E. Akhmetshina, S. Budge, Y. Li, and X. Wu. 2007. "A Survey and Experimental Comparison of Service-Level-Approximation Methods for Nonstationary M(t)/M/s(t) Queueing Systems with Exhaustive Discipline". *INFORMS Journal on Computing* 19(2):201–214.

Ingolfsson, A., F. Campello, X. Wu, and E. Cabral. 2010. "Combining Integer Programming and the Randomization Method to Schedule Employees". *European Journal of Operational Research* 202(1):153–163.

Legros, B., and O. Jouini. 2019. "On the Scheduling of Operations in a Chat Contact Center". *European Journal of Operational Research* 274(1):303–316.

Legros, B., O. Jouini, and G. Koole. 2015. "Adaptive Threshold Policies for Multi-channel Call Centers". *IIE Transactions* 47(4):414–430.

Legros, B., O. Jouini, and G. Koole. 2018. "Blended Call Center with Idling Times During the Call Service". *IISE Transactions* 50(4):279–297.

Luo, J., and J. Zhang. 2013. "Staffing and Control of Instant Messaging Contact Centers". *Operations Research* 61(2):328–343.

Pot, A., S. Bhulai, and G. Koole. 2008. "A Simple Staffing Method for Multiskill Call Centers". *Manufacturing & Service Operations Management* 10(3):421–428.

Tezcan, T., and J. Zhang. 2014. "Routing and Staffing in Customer Service Chat Systems with Impatient Customers". *Operations Research* 62(4):943–956.

AUTHOR BIOGRAPHIES

SIQIAO LI is a PhD Student in the Department of Mathematics at Vrije Universiteit Amsterdam. Her research interests include forecasting, modeling and simulation for improving workforce management in contact center systems. Moreover, also as a PhD Student in the Department of Industrial Engineering at Shanghai Jiaotong University, she has submitted the thesis on the topic of "Capacity Allocation and Patient Scheduling for Radiotherapy Process with Re-entrance and Random Arrivals", applying for the degree of Doctor. Her e-mail address is l.s.q.li@vu.nl.

GER KOOLE is Full Professor in Applied Probability at Vrije Universiteit Amsterdam. His research is centered around the control of queueing systems, and applications of that in various areas, especially call centers, health care and revenue management. His e-mail address is ger.koole@vu.nl.

OUALID JOUINI is Full Professor in Operations Management at CentraleSupélec. His current research interests are in stochastic modeling and service operations management. His main application areas are call centers and healthcare systems. His e-mail address is oualid.jouini@centralesupelec.fr.